**AI CUP 2024 秋季賽**

**根據區域微氣候資料預測發電量競賽**

**報告說明文件**

* 參賽隊伍需詳細說明系統的流程、演算法、工具與外部資源等，特別是創新性(如改了外部資源的哪一部分)。主辦單位會請專業學者、專家仔細審查，若發現分數有問題或方法說明不清之部分，請參賽隊伍補充說明，經發現有違規者，將取消獎項資格。
* 本競賽未禁止使用生成式AI，惟於競賽及撰寫報告過程如有使用生成式  
  AI，請如實揭露，並說明作法。
* 請使用A4紙直式打字，中文字體使用標楷體，英文、數字與符號使用Times New Roman字體。
* 版面設定：邊界上下各2.54CM，左邊3.17CM、右邊3.0CM。
* 字體大小：題目20（粗體）**、**內文12，單行間距。
* 作者聯絡資料表：詳細資料請填寫於附件的「作者聯絡資料表」。
* 須依照以下大綱及內容說明撰寫。不可自訂標題、修改內容順序，或合併段落。但可搭配圖片說明。
* 有意爭取創意獎的隊伍，可參考「[優良報告範本](https://drive.google.com/drive/folders/1_s4S-DVIs0pSkXC1lCyPdFpaWCs3uJvC?usp=share_link)」，惟撰寫格式仍應遵照下方規定。
* 附件空白報告範本可參用。
* **報告大綱與撰寫規定（字數不得低於各段落規定下限）**

壹、環境

字數規定：200~600字。

內容規定：請說明使用的作業系統、語言、套件(函式庫)、預訓練模型、額外資料集等。如使用預訓練模型及額外資料集，請逐一列出來源。

貳、演算方法與模型架構

字數規定：400~1200字。

內容規定：說明演算法設計、模型架構與模型參數，包括可能使用的特殊處理方式。

參、創新性

字數規定：300~1200字。

內容規定：說明演算法之創新性或者修改外部資源的哪一部分。

肆、資料處理

字數規定：300~1500字。

內容規定：說明對資料的處理或擴增的方式，例如對資料可能的刪減、更正或增補。

伍、訓練方式

字數規定：400~1000字

內容規定：說明模型的訓練方法與過程。

陸、分析與結論

字數規定：400~2500字。

內容規定：分析所使用的模型及其成效，簡述未來可能改進的方向。分析必須附圖，可將幾個成功的和失敗的例子附上並說明之。

柒、程式碼(未於**2024/12/09 23:59前**繳程式碼連結者，將失去獲頒獎金/獎狀資格)

內容規定：請在此提供程式碼下載連結並請另外信件附檔，包含資料處理、訓練流程、預測等相關程式碼。程式碼應附README.md檔案交代安裝配置環境，重要模塊輸出/輸入，以讓第三方用戶可以除錯、重新訓練與重現結果。繳交前請確認連結有效且有開啟瀏覽權限，如連結失效視同未交。

捌、使用的外部資源與參考文獻

內容規定：參考文獻請以APA格式為主。

★註1：請確認上述資料與AI CUP報名系統中填寫之內容相同。自2023年起，獎狀製作將依據報名系統中填寫內容為準，有特殊狀況需修正者，請主動於報告繳交期限內來信moe.ai.ncu@gmail.com。，報告繳交截止時間後將不予修改。

★註2：繳交程式碼檔案與報告，請Email至：ailabailab5051@gmail.com，並同時副本至：t\_brain@trendmicro.com與moe.ai.ncu@gmail.com。缺一不可。報告檔名與信件主旨請寫「競賽報告與程式碼/TEAM\_？？？？/根據區域微氣候資料預測發電量競賽」

**AI CUP 2024 秋季賽**

附件

**根據區域微氣候資料預測發電量競賽報告**

隊伍：TEAM\_6894

隊員：沈奕至(隊長)、張鳳愷。

Private leaderboard：635905.5/Rank 23

1. **環境(200~600字)**

**作業系統**：Windows 11 家用版 23H2

**程式語言與套件**：

* **Python** : 3.12.7
* **Jupter套件**：IPython(8.27.0)、ipykernel(6.29.5)、ipywidgets(8.1.2)、jupyter\_client(8.6.0)、jupyter\_core(5.7.2)、jupyter\_server(2.14.1)、jupyterlab(4.2.5)、nbclient(0.8.0)、nbconvert(7.16.4)、nbformat(5.10.4)、notebook(7.2.2)、qtconsole(5.6.0)、traitlets(5.14.3)

在這次的比賽中，我們這組皆用jupyter notebook執行相關程式，包括分析、資料前處理、model訓練等。相較於用vscode .py檔案執行，這種方式比較適合用於做分析、debug。

* **相關import套件：**numpy(1.26.4)、matplotlib(3.9.2)、seaborn(0.13.2)、pandas(2.2.3)、scikit-learn(1.5.2)、xgboost(2.1.2)、scipy(1.14.1)

1. Numpy用於特定的運算及統一data type以利於後面的運算。
2. 我們使用matploblib中的pyplot及seaborn用於數據分析。
3. pandas用於讀取csv檔及部分csv操作。
4. scikit-learn中，我們使用了Ridge、PolynomialFeatures、RobustScaler以及部份metrics套件。
5. xgboost則是使用裡面的XGBRegressor當作模型使用。

**額外資料集(網址附在最後)**：

* **中央氣象局歷史資料(更新頻率 1Hr)**

關於額外資料集，由於Arduino設備的硬體配置可能存在些許差異 (即便硬體品牌相同)而產生不同的特徵數值，所以我們使用了中央氣象局歷史的資料來擴增特徵(2024年1月1日~ 2024年10月31日，更新頻率為1Hr)。[466990 花蓮站]

* **Google Earth**

在競賽的說明pdf檔中，有提到不同偵測站的位置及樓層高度等，我們利用Google Earth找出這些地方，根據當地的一些資訊(ex. 海拔)新增特徵。

1. **演算方法與模型架構(400~1200字)**

**設計架構(圖1.)**：

1. 資料處理: Sunlight(Lux)異常校正、Max\_Sunlight(Lux)校正
2. 預測Sunlight(Lux)
3. 預測Power(mW)

一張含有 文字, 字型, 螢幕擷取畫面, 圖表 的圖片

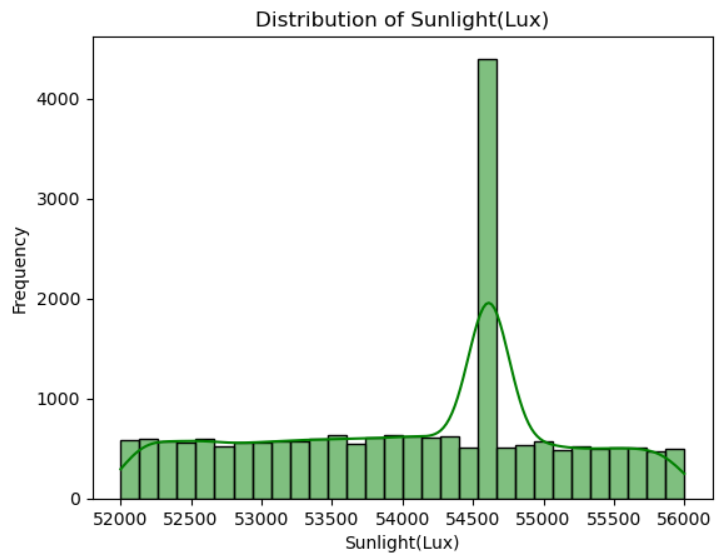
自動產生的描述

<圖1. 流程圖>

**資料處理**：

1. Sunlight(Lux)異常校正(針對數據較穩定的1、2、5、7偵測站)：

在進行數據分析的時候，我們發現Sunlight(Lux)數值為54612.5的資料異常多(圖2)。並且在Sunlight(Lux)-Power(mW)的分布圖中，可以發現在此Sunlight(Lux)數值下的Power(mW)分布範圍幾乎涵蓋了全部範圍(可由表1得知)，推測為異常數值。我們使用既有特徵與Power(mW)等資訊進行修正。



<圖2.Sunlight(Lux)異常值>

參數、模型：

因為選用了數據較為正常的LocationCode進行校正，並且只是為了校正異常值，所以可以使用較強且非線性的Regressor Model。

params= {'colsample\_bytree': 0.95, 'learning\_rate': 0.3, 'max\_depth': 12, 'n\_estimators': 450, 'subsample': 0.95}，使用XGBRegressor(\*\*params)進行校正。

1. Max\_Sunlight(Lux)校正(針對數據較穩定的1、2、5、7偵測站)：

在競賽資料的pdf檔案中有提到，因為硬體上的限制，Sunlight(Lux)欄位最大數值只到達117758.2，所以我們需要將其恢復到應有的數值上。

為了超出數值的上限，我們使用線性的Regressor進行填補，因為線性的Regressor容易受到資料的分布而影響最終的預測結果，所以我們在前面選擇數據較為正常的幾個LocationCode進行訓練、校正。最後再用修正好的資料去校正剩餘的資料。在訓練上，我們也是使用既有特徵與Power(mW)的資料進行預測。

參數、模型：

呈1.，取那些LocationCode為1、2、5、7且Sunlight(Lux)不為最大值(117758.2)的資料，先做PolynomialFeatures(degree=3)再用RoubustScaler做scaling，最後用Ridge預測Sunlight(Lux)。

最後，再將已經校正好的資料(LocationCode 1、2、5、7)作為train\_data使用XGBRegressor將其餘的LocationCode重新做預測、重建。XGBRegressor params = {'colsample\_bytree': 0.95, 'learning\_rate': 0.1, 'max\_depth': 12, 'n\_estimators': 450, 'subsample':0.95}。

**預測Sunlight(Lux)** :

由於test\_data中並沒有Sunlight(Lux)的資訊，所以我們需要先用既有的 資料去預測。

params\_sunlight = {'colsample\_bytree': 0.95, 'learning\_rate': 0.3, 'max\_depth': 12, 'n\_estimators': 350, 'subsample':0.95}， XGBRegressor(\*\*params\_sunlight, objective='reg:squarederror')

**預測Power(mW)** :

將既有的資料與預測完的Sunlight(Lux)再拿去預測power。

params\_power = {'colsample\_bytree': 1.0, 'learning\_rate': 0.1, 'max\_depth': 14, 'n\_estimators': 300}

XGBRegressor(\*\*params\_power, objective='reg:absoluteerror')

1. **創新性(300~1200字)**

在競賽說明文件中有提供每個資料蒐集站的粗略位子，我們能夠經由該表所提供的資訊，透過Google Earth的座標位置查詢，取得更多資訊。在本次比賽中，我們新增了位置、面朝向、高度等資訊。

關於特徵Sunlight(Lux)，競賽說明文件中也有提到其對於Power(mW)具有極高的解釋度，然而裡面仍然有許多噪點需要做修正。由於我們的擴增的氣象特徵是以每小時的頻率做更新，所以多一個基於Power(mW)與其餘特徵去修復、預測的特徵Sunlight(Lux)，可以有效的減少頻率不同步的問題。

除了模型選擇外，為了避免Sunlight模型預測出來的數值有負數的情況，在使用XGBRegressor前，都有把Sunlight進行np.log1p的操作，最後再用np.expm1做復原；在使用Ridge的時候我們並沒有這樣做，因為XGBRegressor較為強大，可以有效的區分絕大多數的數值，但是Ridge預測出來的數值通常有較多的誤差，這會導致最終的結果誤差較大。

與正常的預測模型不同，我們先預測Sunlight(Lux)，再去預測Power(mW)的資訊，由於前面在修正Sunlight(Lux)的過程中，有利用Power(mW)作為一部分特徵去修復，所以此做法可以避免外部資料時間頻率不同所帶來的缺點。

1. **資料處理(300~1500字)**

關於資料前處理的部分，比賽提供的Train Data資料頻率是以分鐘為單位做進行劃分，而最終預測的實際結果則是以十分鐘為單位做劃分。考慮到資料特徵中可能有噪點的問題，我們在資料前處理階段都是以分鐘為單位進行處理，以確保每一筆資料都能得到可靠的數值，避免往後在做資料合併時產生誤差。將所有資料的異常值都處理完畢，再將資料以十分鐘為單位做合併，並對該區間的數據取平均值，作為該時段的平均Power(mW)，方便最後的預測。

由比賽給的Train Dataset及Upload等csv檔可以得知，在Upload中並沒有Train Dataset所給的特徵值，也就是我們需要自己去做填補，亦或是捨棄這些特徵。除此之外，我們在新增特徵的同時，也需要考量Upload的資料是否也能夠做填補。

我們的選擇是將Train Dataset中的所有特徵(除了Sunlight(Lux))都捨棄，並且換成中央氣象局的歷史數據。儘管中央氣象局的資料是以每小時的頻率進行更新，但考量到不同測量站的硬體設備表現可能各不相同(即便是使用相同牌子的硬體設備，仍然可能有些許差異)，這可能導致最終Model的預測受到干擾。

此外，我們也透過Google Earth針對不同偵測站新增了位置、面朝向、高度等資訊。位置區分了東華校區與花蓮氣象站附近的偵測站；面朝向代表太陽能發電板的面朝向；高度是參考該區域的海拔再加上樓層高\*(樓層數-1)，為了方便樓層高我們設為4.2m。

我們在Sunlight(Lux)的修正上，取用特定偵測站的資料(Sunlight(Lux)數據較為穩定的偵測站資料)，將他們裡面的異常值恢復，並且將Sunlight(Lux)卡在最大值上限的數值重新建構(在恢復及建構的訓練特徵中皆有包含Power(mW))。最後，用這些已修復好的資料去重建其餘的資料。

**伍、訓練方式(400~1000字)**

所有的模型訓練均採用8:2的比例分割數據，80%作為train data，20%作為validation data。使用train data所訓練出來的model去predict validation data，選擇表現最好(loss最低)的參數當作最終model的參數使用。除此之外，由於數值的分布不均勻(如圖3所示)，我們使用numpy將數據進行分段，並且使用stratify的方式，確保split後的數據分布能夠更加均勻，降低數據不平衡對模型訓練的影響。

一張含有 文字, 圖表, 螢幕擷取畫面, 繪圖 的圖片

自動產生的描述

<圖3.Sunlight(Lux) Histogram>

在Sunlight(Lux)的修復上，我們使用了Power(mW)作為其中的一個特徵，因為Sunlight(Lux)對於Power(mW)有高解釋性，這正好能夠彌補我們在中央氣象資料特徵擴增上的不足。

在所有與Sunlight(Lux)有關的預測、修正的model中，我們皆使用Mean Squard Error當作loss function。因為在訓練之前，我們選擇的對象的outliers相對較少，這能夠有效的使模型快速收斂。在最終的Power(mW)預測模型中使用Mean Absolute Error當作loss function，單純只是想要與比賽的評分方式相同。

在挑選最佳參數上，比賽的前幾天我們使用RandomizedSearchCV進行參數挑選，過程大約耗時2個小時左右進行參數挑選。然而，最好的成績是在最後一天產生的，當時我們是用類似先前的參數進行訓練，並且透過調整XGBRegressor的max\_depth、n\_estimators選擇最佳loss數值的參數。由於硬體上的限制，我們沒辦法很快速的訓練模型，但是我們仍然推薦使用RandomizedSearchCV或GridSearchCV進行參數挑選，因為他能夠更精確的找到最佳(不會overfitting、loss也較小)的模型參數。

**陸、分析與結論(400~2500字)**

這個部分會講述我們在整個比賽過程中，結構設計、模型選擇上的成效，以及未來可以改進的方向。

**資料前處理、外部資料擴充**：

考量到test data也需要有同樣的特徵，我們將絕大多數的特徵都換成中央氣象局的資料，所以只有特徵Sunlight(Lux)需要做修正。在Sunlight(Lux)的最大值預測上，我們有使用XGBRegressor進行預測過，然而其成效並沒有很好，僅有少數(827/7307，如表1所示)的最大值有成功被預測超過原定的最大值。原因在於此Model是樹狀結構的Regressor，擅長預測既有資料範圍內的數值，但是他在mse(1,556,149)的表現上卻比Ridge的方法好很多。最後我們使用Ridge來進行預測，成功將一半左右(3471/7307)最大值的資料預測到其該有的位置，而其mse為15,922,571，基本上多了XGBRegressor一個檔次，不過我們是為了重建最大值的數值，所以不會影響到其於正常的資料。(以上數據都有在程式中做提供)

從表2可以看出，我們成功將Sunlight(Lux)的數據大部分的異常值都去除，並且有效的集中數據，增加其穩定性。然而越穩定的Sunlight(Lux)資料是否代表更好的Power(mW)？其實不然，在比賽中我們有利用更加細微的特徵，例如：月份，我們找出該月份中較穩定的幾個LocationCode當作我們的train data，去校正其餘的資料，使得最終校正的Sunlight(Lux)更加穩定，但是測試出來的Public Score其實差不多。

|  |  |
| --- | --- |
| LocationCode | Figure |
| 1  原始資料  分布 | 一張含有 螢幕擷取畫面, 文字, 圖表, 繪圖 的圖片  自動產生的描述 |
| 1  使用XGBRegressor修復異常值、重建最大值 |  |
| 2  原始資料  分布 |  |
| 2  使用XGBRegressor修復異常值、重建最大值 |  |
| 5  原始資料  分布 |  |
| 5  使用XGBRegressor修復異常值、重建最大值 |  |

表1. 資料前處理之Sunlight(Lux)-Power(mW)分布圖(使用XGBRegressor)

|  |  |
| --- | --- |
| LocationCode | Figure |
| 1  原始資料  分布 |  |
| 1  XGBRegressor  修正異常值 |  |
| 1  Ridge重建  最大值 |  |
| 2  原始資料  分布 |  |
| 2  XGBRegressor  修正異常值 |  |
| 2  Ridge重建  最大值 |  |
| 5  原始資料  分布 |  |
| 5  XGBRegressor  修正異常值 |  |
| 5  Ridge重建  最大值 |  |
| 原始  整體資料分布 |  |
| 根據1、2、5、7偵測站資料使用  XGBRegressor  進行預測的全體資料分布 |  |

表2. 資料前處理之Sunlight(Lux)-Power(mW)分布圖(使用Ridge)

關於外部資料，我們使用中央氣象局的一小時頻率資料作為氣象特徵的擴增，然而其時間區間較大，可能無法有效表示資料當下實際的氣象狀態。在政府的資料開放平台上也有10分鐘頻率的資料，不過他的期限較短，通常只保留1~3天的資料，不過如果想要更精準的預測太陽能發電量，可以每天都將這些資料儲存下來做為預測的根據。

**預測最終結果-Power(mW)**：

根據我們的模型流程，我們先將Train Data的所有特徵(包括Power(mW))來預測Sunlight(Lux)，最後再去預測Power(mW)。利用兩段式預測label的方式去彌補擴充特徵的不足(時間頻率不同)，然而這種做法是否是正確的仍待確認(由於我們的隊伍在前面幾天一直誤會題目的要求，誤把平均十分鐘的Power(mW)多乘上10，所以導致最後一天才取得較好的成績，因此無法確認此方法是否正確)。

**正確性**：

我們使用中央氣象局的資料去填補預測當下資料的氣候狀況，這樣的模型可能只適合用於修復特定站內的相關Power(mW)資料遺失等問題。(比賽結束前12小時左右，討論區有人提出這個看法，我們才意識到這個問題)

關於預測未來的氣候狀態，我們認為<https://www.windy.com/>、<https://www.ventusky.com/>等衛星網站都能夠有效提供未來的資料及過去的資料，如此一來就能夠在保有氣象特徵的情況下進行預測未來的太陽能發電量了。

**柒、程式碼**

Github連結：<https://github.com/Jackyee1218/2024_AI_CUP>

**捌、使用的外部資源與參考文獻**

1. 中央氣象局（2024）。氣象觀測站資料查詢服務。<https://codis.cwa.gov.tw/StationData>[466990 花蓮站]
2. Google（2024）。Google Earth系統。

<https://www.google.com/intl/zh-TW/earth/about/>

**作者聯絡資料表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 隊伍名稱 | TEAM\_6894 | Private Leaderboard 成績 | 635905.5 | Private Leaderboard 名次 | 23 |
| 身分  （隊長/隊員） | 姓名 | 學校＋系所中文全稱 | 學校＋系所英文中文全稱 | 電話 | E-mail |
| 隊長 | (範例)  沈奕至  Yi-Chih, Shen | 國立中山大學資訊工程學系 | National Sun  Yat-sen University  Department of Computer Science and Engineering | 0983-840-015 | v0983840015＠gmail.com |
| 隊員1 | 張鳳愷  Fong-Kai, Chang | 國立中山大學資訊工程學系 | National Sun  Yat-sen University  Department of Computer Science and Engineering | 0975781219 | 921219kevin@gmail.com |

★註1：請確認上述資料與AI CUP報名系統中填寫之內容相同。自2023年起，獎狀製作將依據報名系統中填寫內容為準，有特殊狀況需修正者，請主動於報告繳交期限內來信moe.ai.ncu@gmail.com。報告繳交截止時間後將不予修改。

★註2：繳交程式碼檔案與報告，請Email至：ailabailab5051@gmail.com，並同時副本至：t\_brain@trendmicro.com與moe.ai.ncu@gmail.com。缺一不可。